11-8-2019

Document Layout Analysis and Recognition Systems

Sai Kosaraju
Kennesaw State University

Document Layout Analysis and Recognition Systems

A Thesis Presented to
The Faculty of the Computer Science Department

by

Sai Chandra Kosaraju

In Partial Fulfillment
of Requirements for the Degree
Master of Science, Computer Science

Kennesaw State University
July 2019

Document Layout Analysis and Recognition Systems

Approved:

_____

Dr. Mingon Kang - Advisor

_____

Dr. Dan Chia-Tien Lo– Department Chair

_____

Dr. Jon Preston - Dean

In presenting this thesis as a partial fulfillment of the requirements for an advanced degree from Kennesaw State University, I agree that the university library shall make it available for inspection and circulation in accordance with its regulations governing materials of this type. I agree that permission to copy from, or to publish, this thesis may be granted by the professor under whose direction it was written, or, in his absence, by the dean of the appropriate school when such copying or publication is solely for scholarly purposes and does not involve potential financial gain. It is understood that any copying from or publication of, this thesis which involves potential financial gain will not be allowed without written permission.

_____

Sai Chandra Kosaraju

Notice To Borrowers

Unpublished theses deposited in the Library of Kennesaw State University must be used only in accordance with the stipulations prescribed by the author in the preceding statement.

The author of this thesis is:

Sai Chandra Kosaraju

1100 S Marietta PKWY,
Marietta, GA 30060

The director of this thesis is:

Dr. Mingon Kang

1100 S Marietta PKWY,
Marietta, GA 30060

Users of this thesis not regularly enrolled as students at Kennesaw State University are required to attest acceptance of the preceding stipulations by signing below. Libraries borrowing this thesis for the use of their patrons are required to see that each user records here the information requested.

ACKNOWLEDGEMENTS

With all respect, I would first like to thank my parents for giving me support I need. Second, I would like to express my very great appreciation to my advisor, Dr. Mingon Kang, for his valuable guidance, sheer support and constant encouragement throughout this entire research. Third, I would like to express my gratitude to DataX Lab members for their self-less support. Furthermore, I would like to thank my brother, sister-in-law and other family members for the motivation, inspiration and moral support.

ABSTRACT

Automatic extraction of relevant knowledge to domain-specific questions from Optical Character Recognition (OCR) documents is critical for developing intelligent systems, such as document search engines, sentiment analysis, and information retrieval, since hands-on knowledge extraction by a domain expert with a large volume of documents is intensive, unscalable, and time-consuming. There have been a number of studies that have automatically extracted relevant knowledge from OCR documents, such as ABBY and Sandford Natural Language Processing (NLP). Despite the progress, there are still limitations yet-to-be solved. For instance, NLP often fails to analyze a large document. In this thesis, we propose a knowledge extraction framework, which takes domain-specific questions as input and provides the most relevant sentence/paragraph to the given questions in the document. Overall, our proposed framework has two phases. First, an OCR document is reconstructed into a semi-structured document (a document with hierarchical structure of (sub)sections and paragraphs). Then, relevant sentence/paragraph for a given question is identified from the reconstructed semi structured document. Specifically, we proposed (1) a method that converts an OCR document into a semi structured document using text attributes such as font size, font height, and boldface (in Chapter 2), (2) an image-based machine learning method that extracts Table of Contents (TOC) to provide an overall structure of the document (in Chapter 3), (3) a document texture-based deep learning method (DoT-Net) that classifies types of blocks such as text, image, and table (in Chapter 4), and (4) a Question & Answer (Q&A) system that retrieves most relevant sentence/paragraph for a domain-specific question. A large number of document intelligent systems can benefit from our

proposed automatic knowledge extraction system to construct a Q&A system for OCR documents. Our Q&A system has applied to extract domain specific information from business contracts at GE Power.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS V

ABSTRACT ..... VI-VII

LIST OF FIGURES 10

LIST OF TABLES 11

1. INTRODUCTION 12
   1.1 OCR Introduction 12
   1.2 Problem Statement 13
   1.3 Contributions 13
   1.4 Thesis Organization 14

2. HIERARCHICAL DOCUMENT RECONSTRUCTION 15
   2.1 OCR Extraction using Structural Attributes 15
      2.1.1 Key Technologies Used 15
      2.1.2 OCR Extraction Algorithm 15
         2.1.2.1 Title Extraction 15
         2.1.2.2 Section or Sub Section Extraction 18
         2.1.2.3 Noise Reduction 19
         2.1.2.4 Document Reconstruction 20
   2.2 Limitations 21

3. TABLE OF CONTENTS RECOGINATION 23
   3.1 Table of Contents Introduction 23

3.2 Related Works 23
3.3 Key Technologies Used 25
3.4 TOC Recogination Algorithm 25
3.4.1 TOC Detection 26
3.4.2 Type Recogination and Parsing 27
3.5 Expirement Setting and Results 29

4. DoT-Net: Document Layout Classification using Texture-based CNN 33
4.1 Document Layout Analysis Introduction 33
4.2 Related Works 33
4.3 DoT-Net Architecture and Methods 36
4.4 Block Detection Algorithm 38
4.5 Dataset 39
4.6 Expirement Results 39

5. Knowledge Extraction 50
5.1 Related Works for Knowledge Extraction 50
5.2 Knowledge Extraction Framework 51
5.2.1 Key Technologies used 53
5.2.2 Knowledge Extraction 55
5.3 Results and Discussion 55
6. CONCLUSION 60
7. REFERENCES 62

LIST OF FIGURES

Figure 2.1: Structural Attributes of Documents 16
Figure 2.2: Extraction of Titles and Page Number 17
Figure 2.3: Section Extraction 19
Figure 2.4: Subsection Extraction 19
Figure 2.5: Noise Removal. 20
Figure 2.6: Dictionary Format of Data 21
Figure 3.1: Overview of the TOC extraction framework. 26
Figure 3.2: One-dimensional Horizontal Projections of TOC and Non-TOC. 27
Figure 3.3: Examples of (a)–(c) Flat TOC and (d)–(f) Hierarchical TOC. 28
Figure 3.4: Example of a Horizontally Diluted Image. 29
Figure 3.5: ROC Curve for TOC Detection 31
Figure 4.1: The architecture of the proposed DoT-Net 36
Figure 4.2: Overview of document analysis algorithm 38
Figure 4.3: ROC curves on tile-wise binary classification. 44
Figure 4.4: Examples of DoT-Net with sample documents 47
Figure 5.1: Flow diagram of knowledge extraction algorithm 52
Figure 5.2: Flow diagram of query expansion algorithm 55
Figure 5.3: Example of expanded query 56
Figure 5.4: Most relevant paragraph to the query 59

LIST OF TABLES

Table 1: Evaluation of TOC Detection 32
Table 2: Parsing of TOC 32
Table 3: Benchmark methods on the experimental settings 43
Table 4: Performance with tile images in one-vs.-rest classification 44
Table 5: Performance with block images in one-vs.-rest classification..... 47
Table 6: Performance with tile images in multiclass classification 49
Table 7: Performance with block images in multiclass classification 49
Table 8: Relevance Ranking of Sections 57
Table 9: Relevance Ranking of Sub Sections 58
Table 10: Relevance Ranking of Sub Sections 58

CHAPTER I

INTRODUCTION

1.1 Introduction to Optical Character Recognition (OCR)

Optical Character Recognition (OCR) is a technique of converting handwritten/scanned texts into machine-encoded text. This technique has been widely used to store physical data records (hard copies) in digital databases. For instance, documents like legal contracts, agreements, and invoices are converted into digital records using OCR and stored in databases. OCR technique improves the efficiency and effectiveness of business by not only improving the business flow but also decreasing the costs. Many advantages OCR documents such as secured, non-editable, and flexible management, makes OCR an essential tool [1,2].

Automatic knowledge extraction from OCR can improve document intelligent systems, such as document search engines and document information retrieval. A number of tools such as ABBY, IBM Watson, and NLP have been developed to automatically extract useful knowledge from OCR documents. However, current approaches (e.g., NLP) often fail on a large document. To overcome this challenge, we developed a knowledge extraction framework which takes user-specific questions as input and provides the most relevant sentence/paragraph for user-specific question. irrespective of document size, our proposed framework achieved promising performance.

The proposed framework has two phases firstly, an OCR document is reconstructed into semi-structured document by proposed document layout analysis algorithms

(Discussed in chapters 2-4). The second phase of the proposed framework takes the user-specific question as input and gives the most relevant sentence/paragraph within a document by using a reconstructed semi-structured document.

1.2 Problem statement

We aim to provide effective solutions to improve the predictive performance of Q&A systems with a large OCR document. To achieve the goal, we will solve the following sub problems:

- to analyze complex OCR document layout by recognizing challenging blocks such as math, tables, etc., from OCR documents,
- to expand the OCR application by converting unstructured OCR document into semi structured document, and
- to efficiently find the most relevant paragraph and sentences.

1.3 Contributions

The main contributions of this thesis are:

- Proposed a new algorithm for Q&A by converting the OCR document into hierarchically structured dictionaries
- Proposed a novel image-based deep learning algorithm to extract the Table of Contents (TOC) information.
- Proposed a novel convolution neural network for identifying different types of elements such as text, image, table, etc.

- Proposed an alternative approach to conventional CNN for texture specific analysis and recognition.
- Developed a framework for extracting the information for PDF (OCR) documents.

1.4 Thesis organization

The thesis is organized as follows. This thesis comprises six chapters: Chapter 1 introduces the thesis and states the problem definition. Chapter 2 discusses our proposed structural attributes-based text mining DLA approach and its limitations. Chapter 3 introduces our proposed machine learning approach of Table of Contents recognition (TOC). Chapter 4 introduces our proposed novel texture-based deep learning architecture DoT- Net to identify the various entities in the document. Chapter 5 describes the proposed Question and Answer system and results. Finally, Chapter 6, is the conclusion.

CHAPTER II
Hierarchical document reconstruction using structural attributes

In this chapter, we briefly introduce our proposed structural attributes-based document reconstruction algorithm.

2.1 OCR extraction using structural attributes

2.1.1 Key technologies used in this algorithm

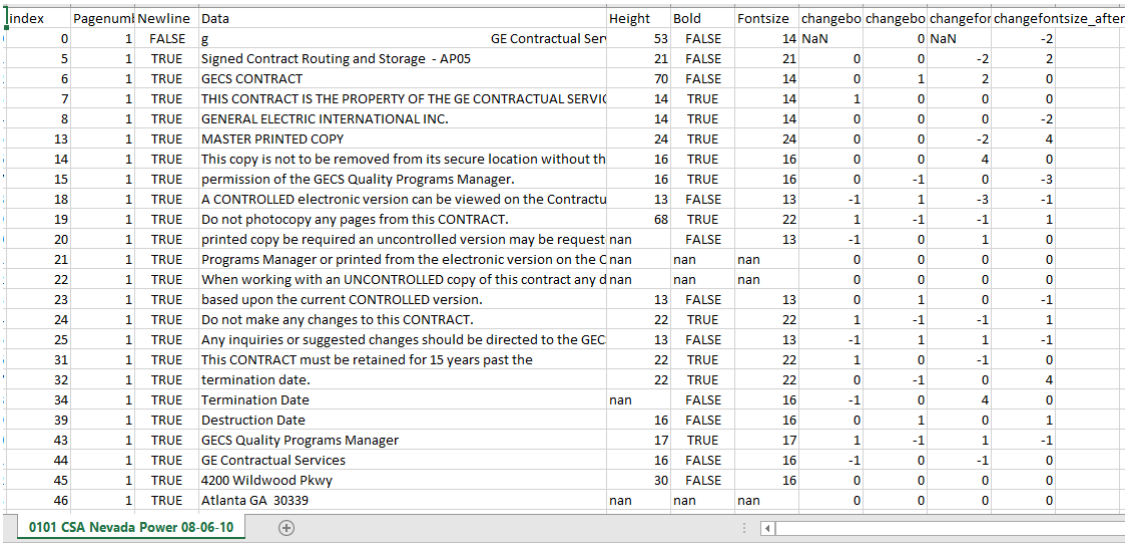
- Tokenization: Tokenization is a technique to split the sentence with special character (.,””, space, etc.)
- Text Tagging: It’s a technique to find text attributes such as font size, bold.
- Regular expression matching: It is technique used to capture regular patterns of text.

2.1.2 OCR extraction algorithm

Main objective of this algorithm is to retain the structure, for example *{title, {subtitle{text}}}*. We extracted titles and substitutes to the text as a key-value pair where the key is the title and value as the text below that. With this algorithm, we can get the OCR document as a semi-structured dictionary (JSON).

2.1.2.1 Title extraction

Restructuring the documents with hierarchical representation, title extraction plays a significant part in the conversion process. In the primary stage of the process, features like font size, page number, boldness of text were determined using text tagging technique to convert given PDF into CSV format. Figure 2.1 shows the attributes of text which are analyzed later for title extraction of contracts.



index	Pagenumb	Newline	Data	Height	Bold	Fontsize	changebo	changebo	changefor	changefontsize_after
0	1	FALSE	g GE Contractual Ser	53	FALSE	14	NaN	0	NaN	-2
5	1	TRUE	Signed Contract Routing and Storage - AP05	21	FALSE	21	0	0	-2	2
6	1	TRUE	GECS CONTRACT	70	FALSE	14	0	1	2	0
7	1	TRUE	THIS CONTRACT IS THE PROPERTY OF THE GE CONTRACTUAL SERVIC	14	TRUE	14	1	0	0	0
8	1	TRUE	GENERAL ELECTRIC INTERNATIONAL INC.	14	TRUE	14	0	0	0	-2
13	1	TRUE	MASTER PRINTED COPY	24	TRUE	24	0	0	-2	4
14	1	TRUE	This copy is not to be removed from its secure location without th	16	TRUE	16	0	0	4	0
15	1	TRUE	permission of the GECS Quality Programs Manager.	16	TRUE	16	0	-1	0	-3
18	1	TRUE	A CONTROLLED electronic version can be viewed on the Contractu	13	FALSE	13	-1	1	-3	-1
19	1	TRUE	Do not photocopy any pages from this CONTRACT.	68	TRUE	22	1	-1	-1	1
20	1	TRUE	printed copy be required an uncontrolled version may be request	nan	FALSE	13	-1	0	1	0
21	1	TRUE	Programs Manager or printed from the electronic version on the C	nan	nan	nan	0	0	0	0
22	1	TRUE	When working with an UNCONTROLLED copy of this contract any d	nan	nan	nan	0	0	0	0
23	1	TRUE	based upon the current CONTROLLED version.	13	FALSE	13	0	1	0	-1
24	1	TRUE	Do not make any changes to this CONTRACT.	22	TRUE	22	1	-1	-1	1
25	1	TRUE	Any inquiries or suggested changes should be directed to the GEC	13	FALSE	13	-1	1	1	-1
31	1	TRUE	This CONTRACT must be retained for 15 years past the	22	TRUE	22	1	0	-1	0
32	1	TRUE	termination date.	22	TRUE	22	0	-1	0	4
34	1	TRUE	Termination Date	nan	FALSE	16	-1	0	4	0
39	1	TRUE	Destruction Date	16	FALSE	16	0	1	0	1
43	1	TRUE	GECS Quality Programs Manager	17	TRUE	17	1	-1	1	-1
44	1	TRUE	GE Contractual Services	16	FALSE	16	-1	0	-1	0
45	1	TRUE	4200 Wildwood Pkwy	30	FALSE	16	0	0	0	0
46	1	TRUE	Atlanta GA 30339	nan	nan	nan	0	0	0	0

0101 CSA Nevada Power 08-06-10

Figure 2.1: Structural Attributes of Documents

In figure 2.1 column 2 indicates the page number, column 3 indicates whether the text is starting in the new line or not, True represents a newline. Column 4 indicates the extracted text. Columns 5,6,7 indicates the text attributes height (height of text block), boldness and, font size respectively. Columns 8,9,10,11 represents the variations of text attributes. Finding variations is an important step for our algorithm. Variations are calculated by subtracting the text attributes of one “i” row with “i-1”.

Variations of font size in the document were investigated and it was assumed that features of titles are bold, higher in font size than before and after the text (sentences). After abstracting all the lines with the expected characteristics for title, we only focused on the lines that include the "string number string" pattern (e.g., "ARTICLE 1 DEFINITION") since this outline is persistent in the title's structures which are detected manually by human. We assumed that each of the contracts follows consistency in the name of title such as comprising by either "Article" or "Section" as title in the contract. The most dominant first string in the "string number string" pattern was considered as first word to start the title name while some predefined words such as "Page", "Figure", and "Table" were excluded. Finally, the text lines were extracted using the starting word (string) that follows a number and another string. Figure 2.2 represents title extraction of one of the contracts.

Index	Page Number	Title
42	7	ARTICLE 1 DEFINITIONS
284	13	ARTICLE 2 CONTRACTOR RESPONSIBILITIES
490	18	ARTICLE 3 OWNER RESPONSIBILITIES
520	18	ARTICLE 4 TERM AND TERMINATION
596	20	ARTICLE 5 PRICE AND PAYMENT TERMS
1211	35	ARTICLE 6 DELIVERY TITLE TRANSFER REPAIR SERV...
1382	39	ARTICLE _7 INSURANCE COVERAGE
1435	41	ARTICLE 8 WARRANTY
1530	43	ARTICLE 9 LIMITATIONS OF LIABILITY
1630	45	ARTICLE 10 DISPUTE RESOLUTION
1681	46	ARTICLE 11 CONFIDENTIAL INFORMATION
1719	47	ARTICLE 12 HEALTH AND SAFETY
1765	48	ARTICLE 13 SUPPLEMENTAL PAYMENT TERMS
1795	49	ARTICLE 14 ASSIGNMENT
1841	50	ARTICLE 15 SITE CONDITIONS AND HAZARDOUS MATE...
1875	51	ARTICLE 16 INDEMNIFICATION

Figure 2.2: Extraction of Titles and Page Number

In the figure 2.2, column 1 represents location of the text within document. Column 2 represents the page number and, column 3 represents the titles.

2.1.2.2 (Sub)section extraction

Once the titles and indices (location of the title) were extracted, text between two consecutive title indices were extracted as section or body of the title. Table of Content (TOC) of a contract also included all the title name and it is assumed that the TOC lies within first five pages of each contract. However, if a contract does not contain TOC or contain after first five pages, page number of TOC can be modified by varying argument in the model. Subsections of a section also extracted using the similar procedure of section extraction where a section was introduced as a document. Figure 2.3 & 2.4 illustrates the section and subsection extraction of a contract document respectively.

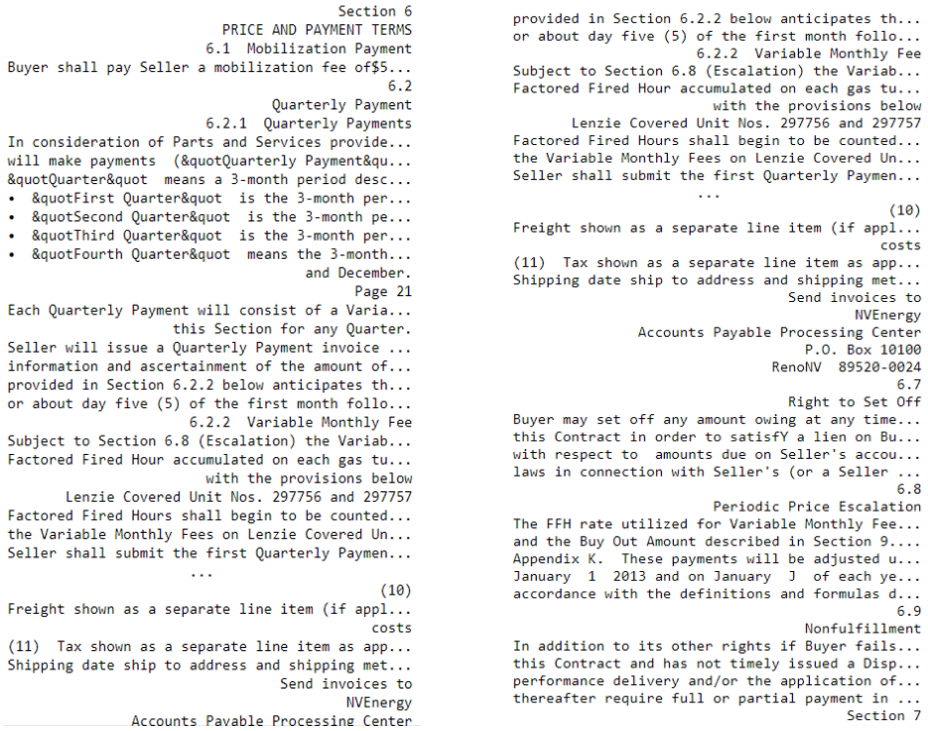


Figure 2.3: Section Extraction

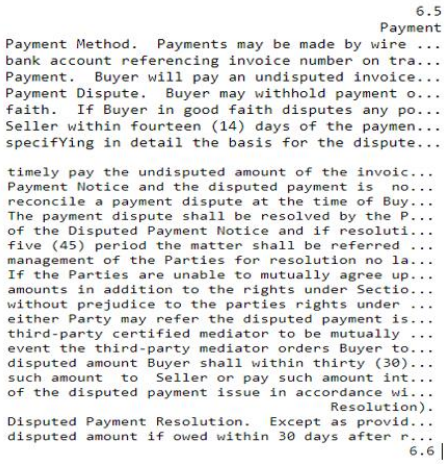


Figure 2.4: Subsection Extraction

2.1.2.3 Noise reduction

Most of the documents has footer and header in watermarks format in every page which causes noise in extracting data from the contract. To build a noise reduction technique, a sample of 20 pages from a contract was taken as sample and converted them into image data for further analysis. In the process, average image of the data was obtained by applying dilution and erosion to the image data. This process resulted in dark areas (Figure 2.5) in footers and headers across multiple pages. Text (footers and headers) in the dark parts were eliminated given the coordinates.

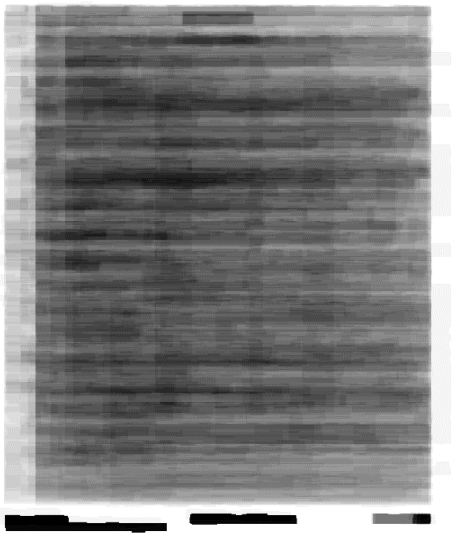
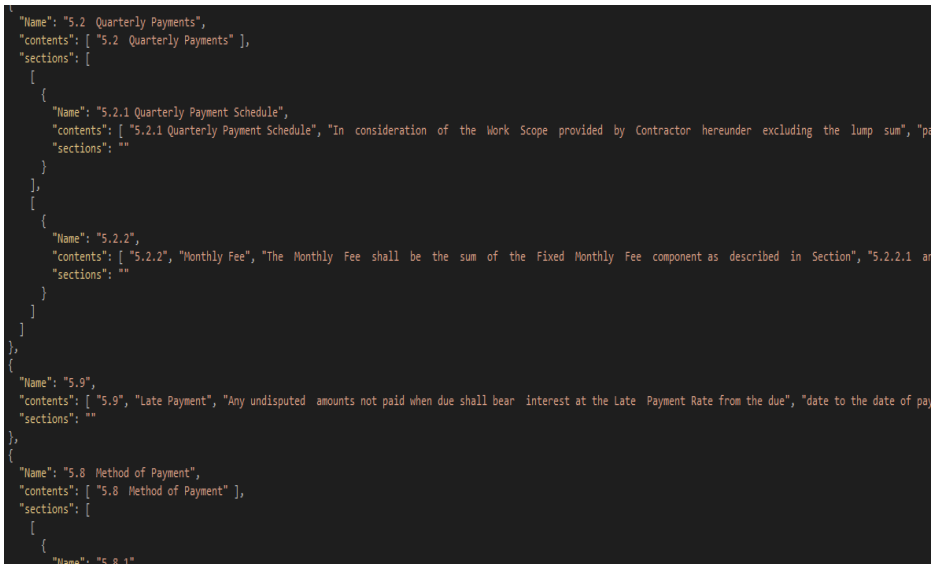


Figure 2.5: Noise Removal

2.1.2.4 Document reconstruction

The extracted title: section/subsection (e.g., Article 5 Price and Payment: Contents of Article 5 Price and Payment) were considered as a key-value pair and converted into dictionary format of JSON file for each of the contract. The hierarchical structure is No-

SQL compatible and facilitates the developing of query/document search system. Figure 2.6 shows an example of JSON format of a contract.



```
{
 "Name": "5.2  Quarterly Payments",
 "contents": [ "5.2  Quarterly Payments" ],
 "sections": [
  [
   {
    "Name": "5.2.1 Quarterly Payment Schedule",
    "contents": [ "5.2.1 Quarterly Payment Schedule", "In  consideration  of  the  Work  Scope  provided  by  Contractor  hereunder  excluding  the  lump  sum", "pa
    "sections": ""
   }
  ],
  [
   {
    "Name": "5.2.2",
    "contents": [ "5.2.2", "Monthly Fee", "The  Monthly  Fee  shall  be  the  sum  of  the  Fixed  Monthly  Fee  component as  described  in  Section", "5.2.2.1  ar
    "sections": ""
   }
  ]
 ]
},
{
 "Name": "5.9",
 "contents": [ "5.9", "Late Payment", "Any undisputed  amounts not paid when due shall bear  interest at the Late  Payment Rate from the due", "date to the date of pay
 "sections": ""
},
{
 "Name": "5.8  Method of Payment",
 "contents": [ "5.8  Method of Payment" ],
 "sections": [
  [
   {
    "Name": "5.8.1"
```


Figure 2.6: Dictionary Format of Data

2.2 Limitations of proposed OCR reconstruction algorithm

The proposed structural attributes-based document reconstruction algorithm has limitations. OCR extraction using conventional text extractors depends on the quality of the text. It is more vulnerable to errors with the average OCR quality.

- **Identification of paragraphs:** The current approach relies on text extractor (for instance, PDF Miner), which makes it difficult to extract the paragraph information.
- **Noise in text:** The current approach lacks to eliminate the noise.

- **Region of Interest:** Document consists of different entities such as table, math, lined. The current approach cannot classify these entities.
- **Table of Contents:** Table of Contents (TOC) plays an important role in the documents. The current approach fails to identify the Table of Contents.

CHAPTER III

Table of Contents recognition using image-based machine Learning

3.1 Table of Contents (TOC) introduction

Table of Contents (TOC) provides an overall structure of a document, so accurate TOC extraction can improve OCR documents analysis efficiency and effectiveness. Most TOC frameworks consist of the three phases: (1) TOC detection, (2) TOC type recognition, and (3) TOC parsing. TOC detection is to locate a "Table of Contents" in a document. Once TOC is located, the type of TOC (flat vs. hierarchical) is determined. Then, TOC entities such as heading title and page number are parsed.

3.2 Related works

Several of studies have been proposed to extract TOC from OCR documents, are broadly drawn into two categories: (1) Rule-based algorithms and (2) Machine learning-based algorithms. Most methods have recognized TOC based on predefined rules with textual features. Naïve rule-based methods typically assume the existence of a keyword, such as "*table of contents"*, or a page number at the end of each line.

Ad-hoc rules were defined on the hypothesis of TOC structure, such as similar spacing and font sizes of text. For instance, keyword matching was considered for TOC detection [32, 34]. The pattern of ordered number elements at the right side of a page was

matched [39]. Predefined mathematical rules were formulated for spaces between both lines and words [33, 38].

A texture pattern of ending with a page number was considered for the rule of entity of TOC [41]. Rule-based TOC recognition methods have shown efficient performance with homogeneous documents that follow strict predefined rules for TOC [37, 40]. However, the rule-based methods often fail with complex and decorative TOC (e.g., table of contents using roman-numerals). Furthermore, the rule-based methods may face problems with OCR documents, because OCR documents often contains or misspelling of words, and noise.

Several machine learning algorithms have been proposed to overcome the limitation of the rule-based algorithms. Textual features including font type, font class, number of contextual/section terms, and line start/end with number, were extracted for machine learning models [39]. Clustering algorithms based on layout features of TOC were proposed to detect and extract TOCs [36, 38]. The layout features included alignment of words, distance of the words, and presence of separators. However, machine learning approaches with textual and layout features of OCR documents are still challenging.

In this chapter, we propose novel image-based machine learning method for TOC recognition. We introduce one-dimensional horizontal projections for efficient and effective TOC detection and propose a simple strategy for TOC recognition using image-based analysis.

3.3 Key technologies used

Machine Learning Technique used in this algorithm:

- **Random Forest:** Random Forest is the ensemble machine learning technique. It operates the multitude of decision tress to overcome the overfitting problem of decision trees.

Other Machine learning techniques used for comparison:

- **Logistic Regression:** logistic regression is used to measure the probability of class.
- **Multi-Layer Perceptron (MLP):** MLP belongs to feed forward network class. MLP has at least three layers input, hidden, and output.
- **Support Vector Machine (SVM):** SVM builds the hyperplane or set of hyperplanes with the help of support vectors, for classification.
- **Decision Tress:** It is technique of splitting the data between nodes based on the Gini index or entropy. It forms a tree like structure for classification.

3.4 TOC recognition algorithm

We propose a TOC recognition framework that can effectively extract TOC entries from OCR documents. In this paper, we mainly focus on developing novel machine learning methods that detect TOC blocks and recognize TOC types (flat vs. hierarchical) using one-dimensional visual features. The overview of our proposed framework is depicted in Figure 3.1. We describe the methods in detail in the following subsections.

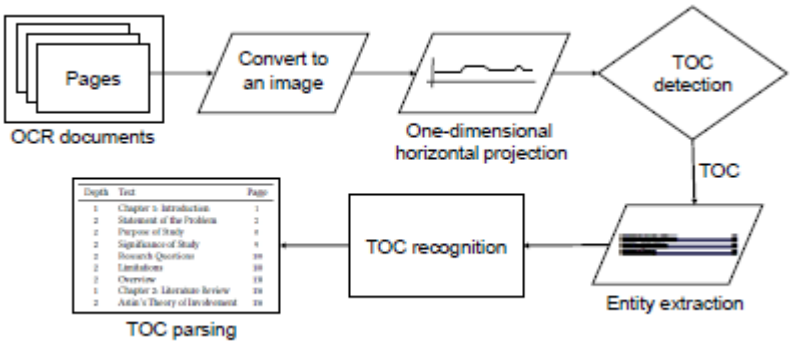


Figure 3.1: Overview of the TOC extraction framework

3.4.1 TOC detection

TOC detection is to locate the table of contents in a document of multiple pages. We assume that TOC spans through an entire page without other major non-TOC blocks (such as texts, tables, and images), and the TOC spans multiple pages. For image-based analysis, the first N-pages of a document are converted into images of a unified size of $\boldsymbol{P} \times \boldsymbol{P}$ pixels (e.g., 256 × 256 pixels), where each page is considered as a single image. Thus, the problem can be considered as a classification task that assigns label of TOC or non-TOC to each page image.

To tackle the classification problem, we propose a machine learning approach using one-dimensional features that represents horizontal projections of a page for efficient TOC detection. The one-dimensional features are simply computed by averaging pixel values on an echo column of a page image. In other words, a feature vector of $1 \times \boldsymbol{P}$ is extracted from a page image of $P \times \boldsymbol{P}$ pixels.

A TOC page image shows a discriminative pattern of the horizontal projection from non-TOC pages, as shown in Figure 3.2. The horizontal projection of TOC shows higher values in the beginning and end sections vs the middle, whereas non-TOC's presents evenly distributed values. Then, the one-dimensional horizontal projections are finally introduced to a classifier for TOC detection.

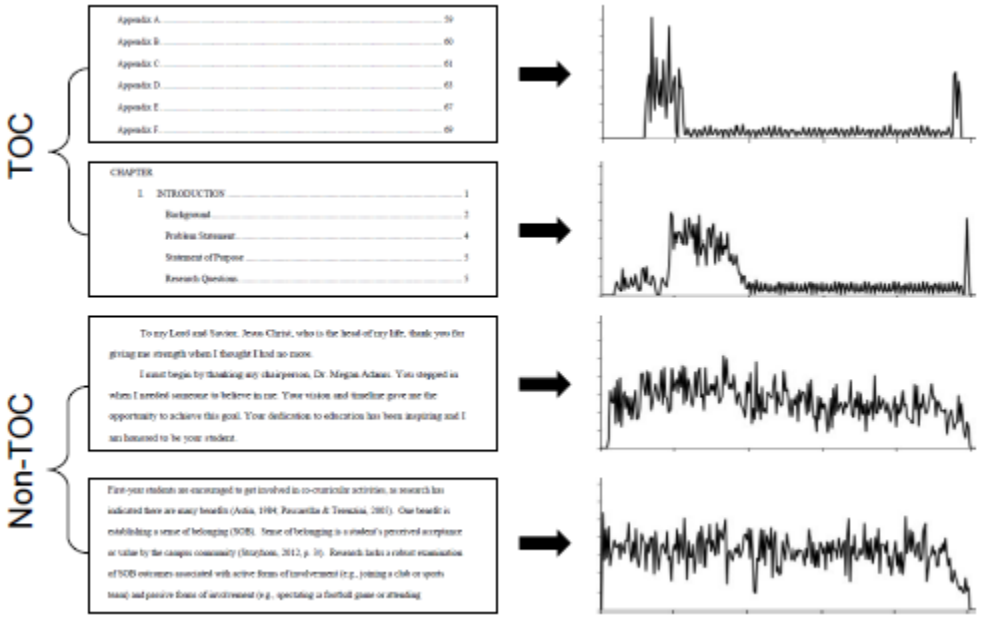


Figure 3.2: One-dimensional Horizontal Projections of TOC and Non-TOC

3.4.2 Type recognition and parsing of TOC

Once a TOC is located, we recognize the type of TOC to effectively extract TOC entities. A TOC belongs to one of the types: flat or hierarchical. A Flat TOC shows that all entries are in a sufficiently similar visual format, and all entries start at the similar horizontal coordinate, as shown in Figure 3.3(a)-(c). While a hierarchical TOC shows entries with

different indentations or font size, with nested entries such as subsection, as shown in Figure 3.3(d)-(f). Therefore, entries in a flat TOC are considered as independent and extracted one by one in order, whereas hierarchical TOC needs hierarchical parsing.

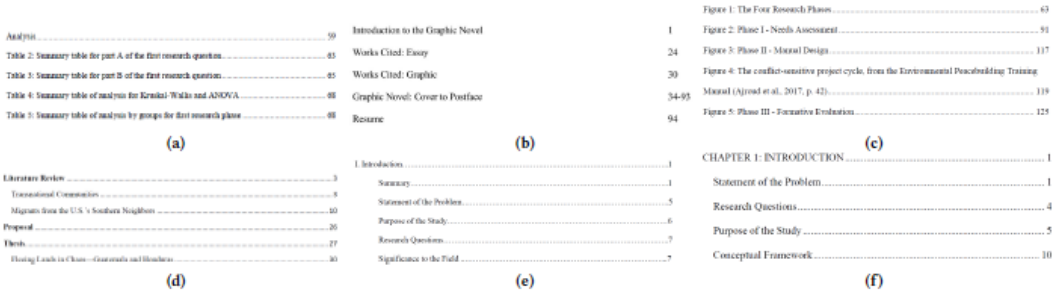


Figure 3.3: Examples of (a)–(c) Flat TOC and (d)–(f) Hierarchical TOC

TOC type recognition is simply implemented by image-based analysis. TOC images are horizontally diluted to obtain connected components (see Figure 3.4). Each connected component corresponds to a TOC entity, which typically contains heading text, separator, and page number. We add coordinates of x and y-axis and a depth to each entity. A depth is determined by comparing the coordinates of top-left corners of the connected components with tolerance pixels (δ). For instance, small differences (<δ) in pixel between the top left corners of the components are considered as the same depth, and depths are increased if other top-left coordinates are observed. Finally, TOC is considered as flat if the depths of all components are the same, otherwise hierarchical.

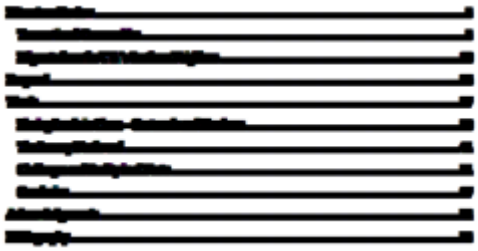


Figure 3.4: Example of a Horizontally Diluted Image

For TOC parsing, we assume that TOC entity follows a regular expression (e.g. "\w+(\W|\s)+\d+$" in Python). The area coordinates of each entity are identified in the previous step, so text of each entity can be separately extracted from the area coordinates. Then, heading title and page number of TOC entity are extracted by the regular expression.

3.5 Experiments setting and results.

Data Description: We conducted experiments to evaluate our proposed method with real documents containing TOC pages. We downloaded 300 PDF files of thesis and dissertation documents which are published since 2015 for master and Ph.D. degrees at Kennesaw State University. The PDF files of thesis and dissertation are available at https://digitalcommons.kennesaw.edu/). Then, we labeled 960 TOC and another 960 non-TOC pages. Note that each document contains several pages of TOC.

Experiments Setting: The dataset was randomly split to three datasets of 560, 140 and 260 samples for training, validation, and test respectively, where the validation data were used for optimizing hyper-parameters. Each page was converted to a single image of 256 × 256 pixels. Then, the one-dimensional horizontal projections were computed. We

repeated the experiments ten times for model stability. We evaluated the performance with several machine learning classifiers that input our proposed one-dimensional projections: (1) Multi-layer Perceptron (MLP), (2) Support vector machine (SVM), (3) Logistic regression, and (4) Random Forest.

MLP was with 16 nodes of one hidden layer and relu activation function. SVM was with radial basis function (RBF) kernel, where kernel coefficient (γ) was set to 0.1 and L-2 regularization parameter (C) was 0.01. Logistic regression with L-2 regularization was used with C=0.02. Random Forest classifier with N−estimators (number of trees) = 20 by considering Gini index is trained with one dimensional horizontal projections. Furthermore, we compared the performance with a decision tree, which was the latest machine learning method for TOC detection.

We evaluated the methods with accuracy and F1 score. Confusion matrices were computed with the definition:

- True Positive (TP): correctly identified TOC as TOC,
- False Positive (FP): incorrectly identified non-TOC as TOC,
- False Negative (FN): incorrectly identified TOC as non-TOC,
- True Negative (TN): correctly identified non-TOC as non-TOC.

Then, accuracy and F1 score were calculated by (TP + TN) / (TP + FP + FN + TN) and 2(PPV × TPR) / (PPV + TPR) respectively, where TPR = TP /(TP+FN) and PPV = TP /(TP+FP).

The experimental results are shown in Table 1. Random forest with the proposed one-dimensional projection features outperformed others, where Random forest showed the highest scores on both accuracy and F1 score of 0.87 ~ 0.011 and 0.86 ~ 0.008 respectively. Moreover, Receiver Operating Characteristic (ROC) curve was plotted over the thresholds to examine the trade-off between True Positive Rate (TPR = TP/(TP + FN)) and False Positive Rate (FPR = FP/(FP +TN)). The plot of ROC curve in Figure 3.5 shows the outstanding performance of with the highest area under the curve. Table 2 illustrates an example of TOC entities recognized by our proposed methods with hierarchical TOC. TOC entities include depth, heading text, and page number. Hierarchical TOC shows various depths that represent hierarchy of the structures. The experiments were implemented in Python. OpenCV library in Python was used for image processing, and PDFMiner library was used for TOC text parsing.

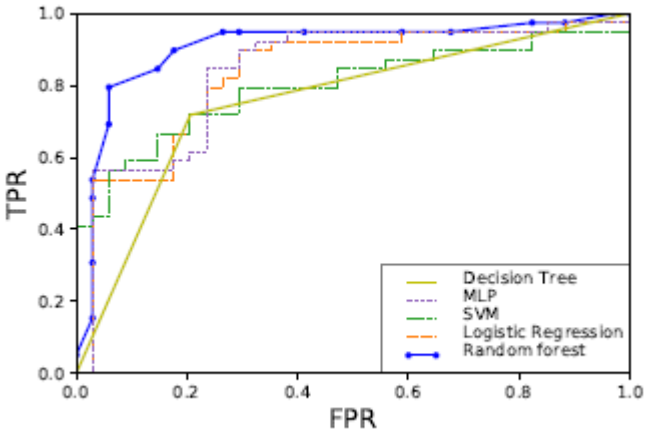


Figure 3.5: ROC Curve for TOC Detection

Table 1: Evaluation of TOC Detection

Methods	Accuracy	F1 score
Decision tree	0.72 ± 0.017	0.73 ± 0.012
SVM	0.61 ± 0.007	0.71 ± 0.004
MLP	0.63 ± 0.013	0.64 ± 0.011
Logistic regression	0.75 ± 0.009	0.79 ± 0.004
Random forest	0.87 ± 0.011	0.86 ± 0.008

Table 2: Parsing of TOC: Hierarchical TOC

Depth	Text	Page
1	Chapter 1: Introduction	1
2	Statement of the Problem	2
2	Purpose of Study	8
2	Significance of Study	9
2	Research Question	10
1	Chapter 2: Literature Review	15
3	Sense of belonging factors	23

CHAPTER IV

DoT-Net: Document layout classification using texture-based CNN

4.1 Document layout analysis (DLA) introduction

Document Layout Analysis (DLA) is a segmentation process that decomposes a scanned document image into its blocks of interest and classifies them, e.g. text, image, table, mathematical expression, and line-diagram [42]. DLA leads to a large number of applications, such as information retrieval, machine translation, Optical Character Recognition (OCR) systems [43], and structured data extraction from documents [44]. However, classification of document blocks in DLA is challenging, due to variation of block locations, inter- and intra- class variability, and background noise. Mainly consists of three procedures: (1) detecting document blocks of interest, (2) extracting features, and (3) classifying the blocks.

We used an existing block detection algorithm to detect blocks. Once blocks are detected a novel CNN texture-based approach was built, for extracting features and classifying the blocks. In this chapter we mainly focus on extracting features and classifying blocks.

4.2 Related works

Traditional block detection methods of top-down, bottom-up, and hybrid approaches have been used to localize document blocks [45]. Then, features are extracted from the blocks by using block-based, pixel-based, or connected component-based techniques [42]. In particular, pixel-based features include entropy, gradient shapes, and contrasts, whereas texture-based features contain size, shape, stroke width, and positions of the blocks. The extracted features are then introduced into a machine learning algorithm for classifying the document blocks. In this study, we focus on classification of document blocks where localized document blocks are given.

A number of machine learning algorithms have been applied for document layout classification with features that describe characteristics of document blocks. Gradient shape features were generated to represent textual patterns and introduced to a Support Vector Machine (SVM) classifier for classifying text blocks from non-text blocks [47]. A Multilayer Perceptron (MLP) was trained with Histogram of Oriented Gradients (HOG) descriptor to classify text and non-text document blocks [46]. An adaptive boosting (Adaboost) decision tree was applied for classification between text and non-text regions using features extracted by the connected component approach [49].

Recently, deep learning has been widely explored in document layout classification. A feed forward neural network was trained with textural and statistical features extracted by processing a mask function across document images for the text vs. non-text classification [50]. A fast-Convolutional Neutral Network (CNN) based document layout analysis was introduced, where two one-dimensional projection of images were

considered to train the model [51]. To identify complex document layouts, a CNN architecture that learns a hierarchy of features from a raw image was proposed for the document image classification [52]. A Deep CNN architecture was applied for classification, where CNNs were extensively used for both feature extraction and model training process [53].

Most DLA studies have been mainly focused on a binary classification between text and non-text blocks. Meanwhile, non-text types of blocks such as table, image, mathematical expression, and line-diagram also play an important role in applications of DLA. However, there has been a little published research for classifying specific non-text types of blocks. For instance, a gradient boosted decision tree-based classification model was adopted to recognize layout tables and extract encoded knowledge from the tables [54]. Therefore, multiclass classification approaches may increase the efficiency and the scope of the document layout analysis.

In this chapter, we propose a document texture-based CNN (DoT-Net), which can effectively and simultaneously classify multiple classes of document blocks (see Fig. 1). Our main contributions are: (1) adopting a dilated convolutional layer replacing all convolutional layers for the texture-based analysis, (2) automatic feature extraction via a deep learning model rather than using explicitly predefined features, and (3) extending to multiclass classification whereas previous methods have typically focused on binary classification of text vs. nontext.

4.3 DoT-Net architecture and methods

DoT-Net enhances a deep learning architecture for document layout classification. DoT-Net adopts dilated convolutional layers to extract texture patterns from document blocks, which tackle the drawbacks of conventional CNN.

Architecture:

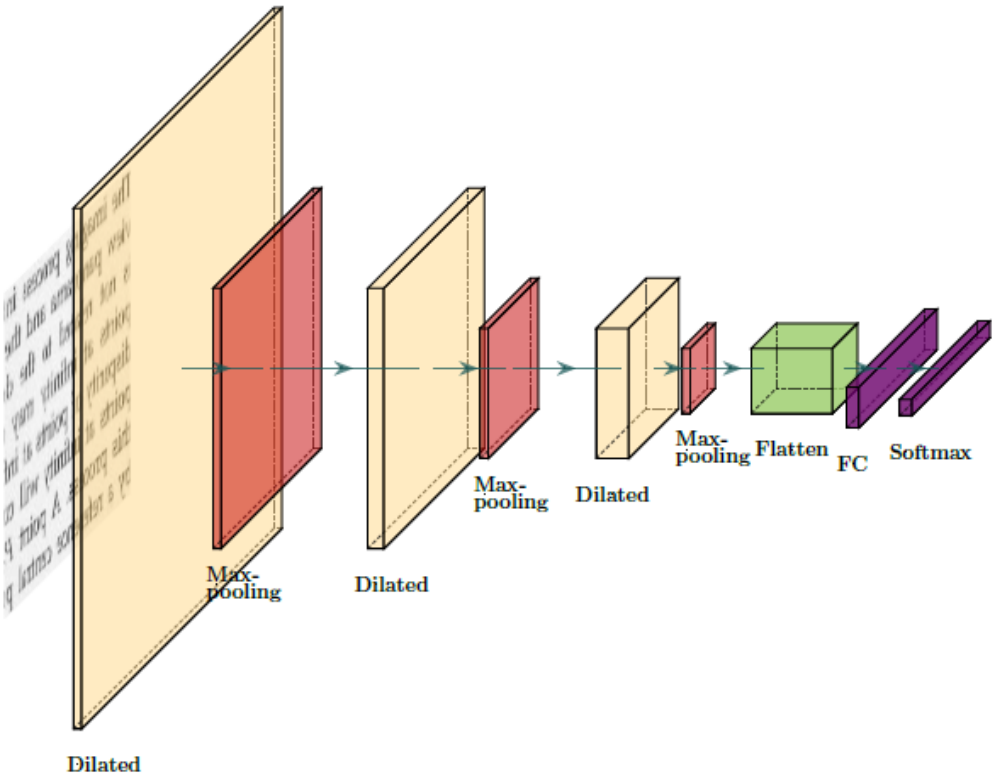


Figure 4.1: The architecture of the proposed DoT-Net

DoT-Net consists of an input layer, three dilated convolutional layers, a flatten layer, a fully connected layer, and an output layer, as shown in Figure 4.1.

Importance and functionality of DoT-Net: Dilated convolutional layers have been widely used for object localization, as alternative of conventional convolutional layers [55], [56]. Dilated convolutional layers enlarge field-of view (texture) of filters without loss of spatial information [57]. In dilated convolutional layers, the numbers of parameters do not increase while enlarging a kernel size, which makes model training computationally efficient. Moreover, dilated convolutional layers trade off context assimilation against computational time [45].

Dilated convolutional layers can capture texture patterns from an image. DoT-Net fully takes the advantage of dilated convolutional layers for texture-based analysis in document layout classification. In DoT-Net, conventional convolutional layers are replaced by dilated convolutional layers, where each dilated convolutional layer is followed by a max pooling layer to control layer sizes in between two dilated layers. Without the max pooling layer, the size of the following layer would increase due to enlarged kernels of dilated convolution. Thus, most studies have used dilated convolutional layers as a deconvolutional layer or the last layer following convolutional layers to localize objects in images [58]– [63]. To the best of our knowledge, no studies have adopted dilated convolutional layers while replacing all convolutional layers. Note that most related works in DLA takes the input of features extracted from blocks, whereas DoT-Net classifies tile images directly by automatically learning distinguishable texture patterns of document blocks.

The overall procedures of document layout analysis are as follows. Given a document that has gone through OCR, document blocks are first localized. Then, tile images are generated by sliding a sub-window across a block. Each tile image is classified by DoT-Net. Finally, the document block is classified by majority voting. The procedure is illustrated in Figure 4.2.

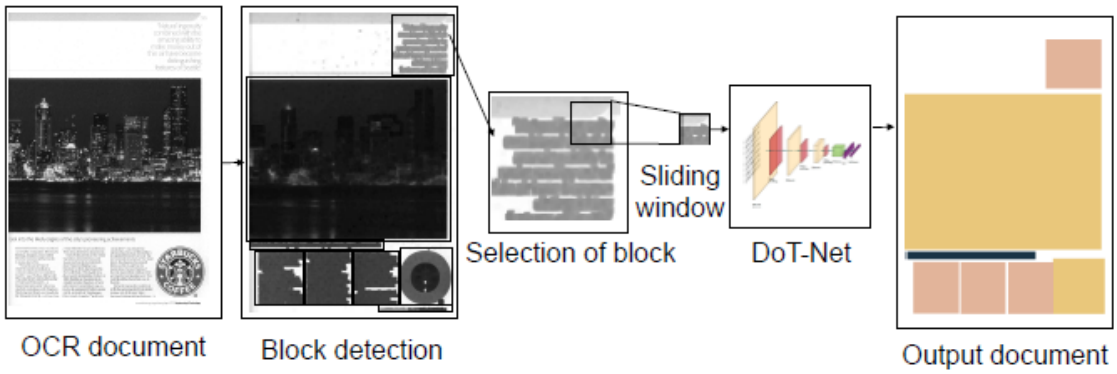


Figure 4.2: Overview of document analysis algorithm.

4.4 Block detection algorithm

Bottom-up approach is technique is widely used technique in number of applications [42]. Whereas, as few applications used top-down and hybrid approaches. For this work we used modified bottom-up-approach [65].

- Image Dilation: Image dilation is a morphology operation which expands the pixels within the image.
- Connected Components: The maximal connected elements are called as connected components.

We recursively dilated image until number of connected components in the converge. Once the image is converged each connected component is considered as one block.

4.5 Dataset

We conducted extensive experiments to assess the proposed method, DoT-Net. We used ICDAR document layout analysis dataset [63] that consists of more than 400 annotated documents. The dataset includes fourteen annotated blocks such as text, table, and images. Among them, we considered the five major block categories of text (n = 1, 432), image (248), table (119), math (91), and line-diagram (82). We downloaded the dataset from: https://www.primaresearch.org/dataset.

4.6 Experimental results

We evaluated the performance of our multiclass classifier with existing cutting-edge methods with the following two experiment settings: (1) one-vs.-rest and (2) multiclass classification using either tile or block images extracted from PDF documents. Since most related studies have focused on a binary classification problem between text and non-text blocks, the multiclass classification problem was converted into one vs.- rest classification problems for the comparison. For each one-vs.-rest classifier, we randomly selected 80 blocks on a positive class and 20 on each of the other classes, so that totally 80 blocks are from the remaining classes. For the multiclass classification experiments, we randomly selected 80 blocks per class.

Moreover, we considered two different types of input data: (1) blocks and (2) tiles of a block. A block was introduced to benchmark methods in block-wise experiments, whereas a tile of a block was an input in tile-wise experiments. We measured the

performance based on the input. Document blocks were given from the labeled dataset. Given 80 document blocks per class, we generated tiles by sliding a small window of 100 × 100 pixels and a stride of 30. Finally, 15,000 tiles of 100 × 100 pixels on average were available for per class.

Accuracy, F1 score, and Area Under Curve (AUC) were measured to assess the performance of the methods. We obtained true positive (TP), false positive (FP), true negative (TN), and false negative (FN) on each experiment. Accuracy is denoted by the overall prediction accuracy, which was measured by (TP + TN) / (TP + FP + TN + FN). F1 score was calculated by 2 × (precision × recall) / (precision + recall), where precision = TP / (TP + FP) and recall = TP / (TP + FN). Receiver Operating Characteristic (ROC) curves were traced over various thresholds to examine the trade-off between True Positive Rate (TPR = TP / (TP + FN)) and False Positive Rate (FPR = FP / (FP + TN)). Then, an AUC was computed by the area under the ROC curve.

DoT-Net was implemented by Keras with TensorFlow backend. We set the kernel size to 3 × 3 and dilation rate to 2. A TanH activation function was used for dilated convolution layers with 50 filters. The max pooling layer of size 2×2 with dropout of 0.1 between each max pooling layer and dilated convolutional layer was used. The fully connected layer with 50 nodes and the softmax layer with 5 nodes were considered. We also applied minibatch training, where each minibatch size was 32. Two hyperparameters, learning rate and weight decay, were optimized automatically by grid search, to minimize the error in validation data for each experiment.

We compared the performance of DoT-Net with five document layout methods of both binary and multiclass classifiers. The benchmark classifiers were included: Feed Forward Networks (FFN) [8], Fast One-Dimensional CNN (F1DCNN) [9], Support Vector Machine with Gradient shape features (GSVM) [5], Multilayer Perceptron with HOG features (HOGMLP) [6], and conventional CNN (CNN) [23]. We used 5-fold cross-validation, where 20% of the training data was used as validation data for optimizing hyper-parameters for each experiment. All experiments were repeated ten times for model reproducibility

Experimental settings for the five benchmarks are as follows:

1) **Feed Forward Networks (FFN):** A non-overlapping mask of size 5 × 5 across a resized input block (256 × 256 pixels) generated six statistical features of median, mode, entropy, contrast, energy, and homogeneity [50]. The features were input to the feed forward network that consists of an input layer and a hidden layer with 7 nodes and ReLu activations.

2) **Fast One-Dimensional CNN (F1DCNN):** Fast one-dimensional CNN was proposed to overcome the computational expense of conventional CNNs [51]. Vertical and horizontal projections of one-dimensional array were used as an input. The architecture follows two individual 1DCNN tracks, each of which contains sequence of three one-dimensional convolutional layers with kernel size of 3 ×1. Each convolutional layer followed by max pooling with 2 pixels and 0.1 dropout.

3) **Support Vector Machine with Gradient shape features (GSVM):** GSVM is a block-based classifier. Gradient shape features extracted from a block were introduced to Support vector machine with RBF kernel [47].

4) **Multilayer perceptron classifier with HOG features (HOGMLP):** HOGMLP is a block-based classifier, where Histogram of gradient shape features of a block (HOG features) were input to Multilayer perceptron [48].

5) **Conventional CNN:** We also included the conventional CNN that is LeNet-5 [64], as a baseline method to compare with DoT-Net, although there is no study that directly used CNN for document layout classification. Three convolutional layers with 3 × 3 kernel size, 50 filers and TanH activations were used for optimal performance. Each convolutional layer was followed by max pooling layer with 2 × 2-pixel kernel. Dropout of 0.3 was used between max pooling and convolution layers for the optimization. A 100 × 100 2D image tile was introduced to the CNN model (Same as DoT-Net).

Note: All benchmark methods, except CNN, were initially developed for the binary classifier of text vs. non-text blocks. However, the methods were easily extended for a multiclass classifier. Deep learning-based methods (FFN, F1DCNN, HOGMLP, and CNN) were simply extended for multiclass classifiers by adding five nodes in the output

layer with softmax activation. GSVM was also simply extended with a conventional multiclass SVM model. The optimal hyperparameters were determined by grid search with validation data for all of the benchmark methods.

For block-wise experiments, the tile-wise methods of FFN, F1DCNN, CNN, and DoT-Net first classified the tiles generated from a block, and then made the final decision by majority vote. The block-wise classifiers of GSVM and HOGMLP were not considered in the tile-wise experiments. The benchmark methods considered on each experiment setting are listed in Table 3.

TABLE 3: Benchmark methods on the experimental settings

	Tiles	Blocks
One-vs.-rest	FFN, F1DCNN, CNN, DoT-Net (Results in Table 4)	GSVM, HOGMLP, FFN, F1DCNN, CNN, DoT-Net (Results in Table 5)
Multiclass	FFN, F1DCNN, CNN, DoT-Net (Results in Table 6)	GSVM, HOGMLP, FFN, F1DCNN, CNN, DoT-Net (Results in Table 7)

The experimental results of tile-wise and block-wise binary classification (one- vs. -rest) are summarized in Table 4 and Table 5, respectively. For tile-wise binary classifications (Table II), DoT-Net obtained the highest accuracy and F1 score across the five classes. CNN achieved slightly higher AUCs (0.998 ± 0.013 and 0.994 ± 0.012) in the classes text and image than DoT-Net (0.998 ± 0.015 and 0.992 ± 0.018), while DoT-Net shows the outstanding AUCs in the rest classes (i.e. table, math, and line-diagram). In Fig. 4.3, the corresponding ROC curves demonstrate that the performances

of CNN have dramatically dropped especially in the classes math and line diagram, whereas DoT-Net shows an outstanding AUC with the five binary classifications on average. Similarly, DoT-Net outperformed the five benchmarks in most classes except text in the block-wise binary classification (see Table 5). In the text-vs.-rest classification, CNN obtained the highest accuracy (0.981 ± 0.014) and AUC (0.997 ± 0.030), while DoT-Net obtained the highest F1 score (0.981 ± 0.022). In most cases, DoT-Net shows a robust predictive performance with the least standard errors.

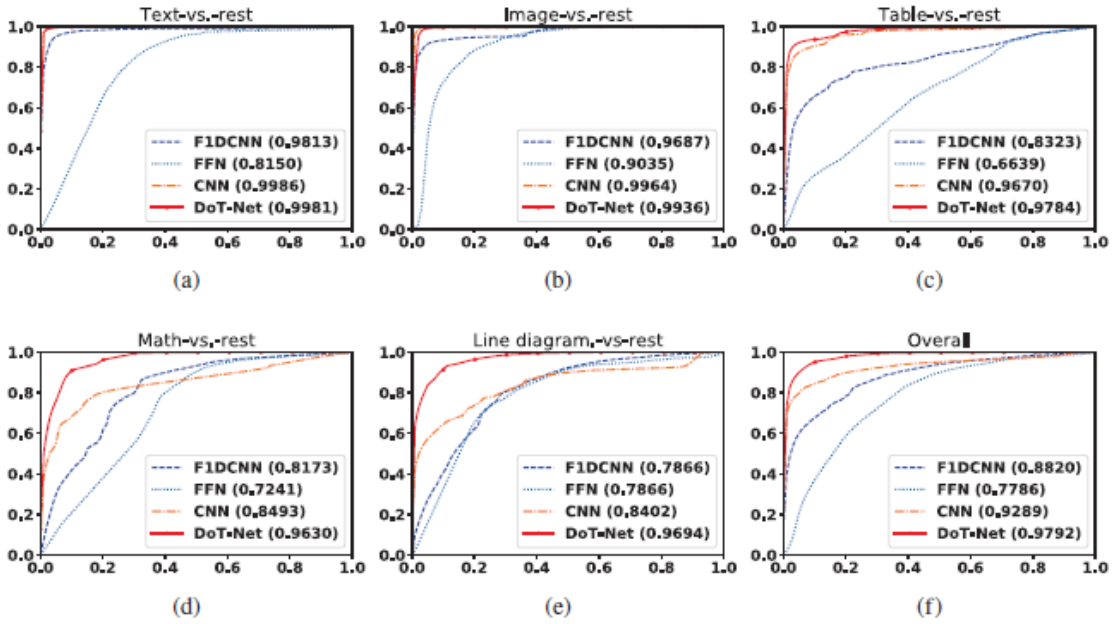


Fig. 4.3: ROC curves on tile-wise binary classification. (a) text-vs.-rest, (b) image-vs.-rest, (c) table-vs.-rest, (d) math-vs.-rest, (e) line diagram-vs.-rest, and (f) averaged overall ROC curves.

TABLE 4: Performance with tile images in one-vs.-rest classification

	Methods	**Accuracy**	**F1 score**	**AUC**
	F1DCNN	0.925 (0.047)	0.884 (0.052)	0.981 (0.014)
	FFN	0.841 (0.017)	0.850 (0.027)	0.903 (0.012)
	CNN	0.945 (0.016)	0.948 (0.014)	0.998 (0.013)

Text	**DoT-Net**	0.981 (0.021)	0.982 (0.013)	0.998 (0.015)
Image	**F1DCNN**	0.905 (0.036)	0.908 (0.041)	0.968 (0.030)
	FFN	0.841 (0.017)	0.850 (0.027)	0.903 (0.012)
	CNN	0.937 (0.009)	0.919 (0.011)	0.994 (0.012)
	DoT-Net	0.970 (0.019)	0.971 (0.017)	0.992 (0.018)
Table	**F1DCNN**	0.773 (0.046)	0.700 (0.051)	0.832 (0.066)
	FFN	0.596 (0.022)	0.644 (0.013)	0.663 (0.018)
	CNN	0.879 (0.028)	0.893 (0.020)	0.965 (0.038)
	DoT-Net	0.917 (0.022)	0.919 (0.018)	0.978 (0.025)
Math	**F1DCNN**	0.825 (0.053)	0.870 (0.037)	0.817 (0.041)
	FFN	0.705 (0.031)	0.748 (0.027)	0.724 (0.028)
	CNN	0.806 (0.015)	0.845 (0.013)	0.849 (0.017)
	DoT-Net	0.900 (0.037)	0.898 (0.026)	0.963 (0.017)
Line-diag.	**F1DCNN**	0.769 (0.027)	0.803 (0.052)	0.810 (0.033)
	FFN	0.737 (0.010)	0.753 (0.028)	0.786 (0.012)
	CNN	0.769 (0.027)	0.718 (0.042)	0.840 (0.048)
	DoT-Net	0.903 (0.024)	0.901 (0.027)	0.969 (0.010)

More importantly, DoT-Net obtained the best performance among three measurements in the tile-wise multiclass classification in Table 6 and block-wise multiclass classification in Table 7. The proposed method Dot-Net appeared the best performance on the three measurements in both the tile-wise and block-wise multiclass classification. Especially, DoT-Net remarkably improved the model performance around 10% comparing to the second-ranked model, F1DCNN in the block wise multiclass classification (see Table 7). The results show that DoT-Net is a robust and accurate classifier for both binary and multiclass problems.

Furthermore, we applied DoT-Net for multiple documents which were not included in the training phase. We used the model with best F1-score. For document block detection, each page in a document was converted into a gray-scale image. Document blocks were detected by a modified bottom-up approach [24], and then tile images were generated by sliding a window of 100×100 pixels with stride of 30. Both correctly classified blocks (Fig. 4.4a–4.4c) and incorrectly classified blocks (Fig. 4.4d–4.4f) are shown, where left side figures are the original documents and the right-side figure illustrated the classification results with different colors (see color legend in Fig. 4.1). In Fig. 4.4d, left bottom block (in red) of the document contains XML code. Dot-Net classified the block as math, whereas the ground truth was text. The block may be misclassified, due to the lack of enough numbers of training data of codes and its similar texture patterns with math (e.g., indentation and special character). In Fig. 4.4e, Dot-Net misclassified the table block in the left-bottom side of the document as math. The misclassification may be caused by the multiple numerical values that are not

differentiated by lines in the table unlike most regular formatted tables have. In Fig. 4.4f, the block of text was classified as table due to the noise in the background.

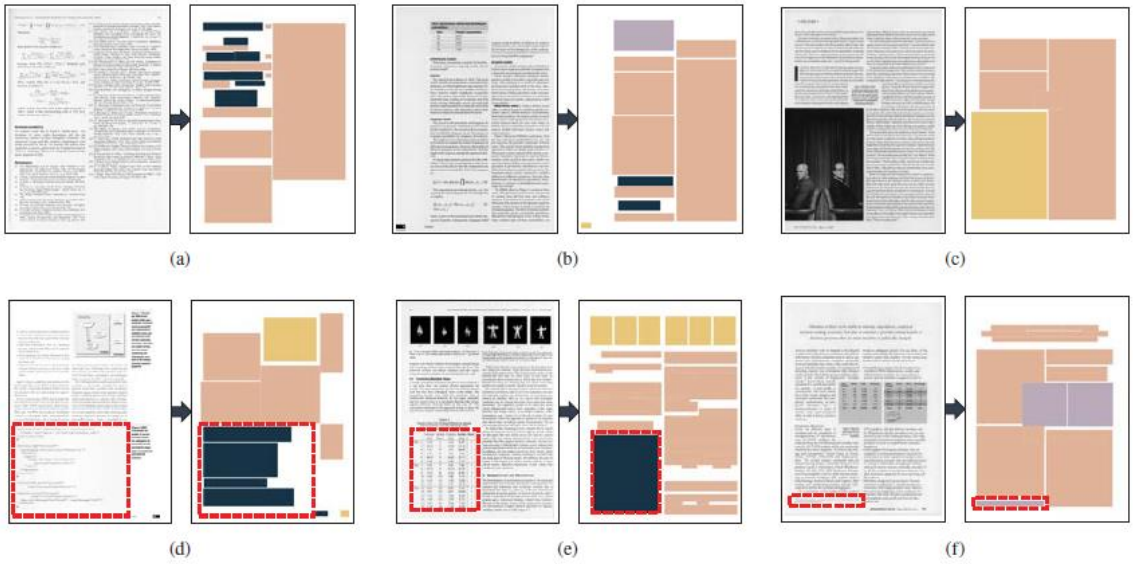


Fig. 4.4: Examples of DoT-Net with sample documents. (a) –(c) Correctly classified blocks and (d) – (f) incorrectly classified blocks. Left side figures are the original documents and the right side figure illustrated the classification results with different colors (see color legend in Fig. 1).

TABLE 5: Performance with block images in one-vs.-rest classification

	Methods	Accuracy	F1 score	AUC
Text	**F1DCNN**	0.936 (0.044)	0.941 (0.032)	0.940 (0.039)
	FFN	0.809 (0.021)	0.844 (0.028)	0.835 (0.043)
	CNN	0.981 (0.014)	0.976 (0.012)	0.997 (0.030)
	GSVM	0.821 (0.015)	0.831 (0.029)	0.815 (0.010)
	HOGMLP	0.783 (0.039)	0.762 (0.011)	0.790 (0.045)
	DoT-Net	0.978 (0.025)	0.981 (0.022)	0.991 (0.024)

Image	**F1DCNN**	0.912 (0.029)	0.905 (0.031)	0.941 (0.010)
	FFN	0.893 (0.021)	0.881 (0.024)	0.943 (0.032)
	CNN	0.943 (0.017)	0.932 (0.020)	0.966 (0.018)
	GSVM	0.869 (0.052)	0.862 (0.061)	0.872 (0.056)
	HOGMLP	0.851 (0.043)	0.825 (0.059)	0.863 (0.039)
	DoT-Net	0.974 (0.017)	0.963 (0.027)	0.971 (0.019)
Table	**F1DCNN**	0.813 (0.037)	0.845 (0.046)	0.872 (0.013)
	FFN	0.712 (0.017)	0.610 (0.036)	0.887 (0.032)
	CNN	0.843 (0.057)	0.829 (0.042)	0.861 (0.035)
	GSVM	0.581 (0.012)	0.771 (0.016)	0.482 (0.012)
	HOGMLP	0.682 (0.018)	0.773 (0.019)	0.635 (0.021)
	DoT-Net	0.911 (0.028)	0.877 (0.019)	0.927 (0.038)
Math	**F1DCNN**	0.748 (0.031)	0.763 (0.024)	0.817 (0.017)
	FFN	0.599 (0.033)	0.611 (0.021)	0.556 (0.033)
	CNN	0.612 (0.018)	0.561 (0.031)	0.646 (0.046)
	GSVM	0.633 (0.050)	0.549 (0.053)	0.650 (0.056)
	HOGMLP	0.689 (0.060)	0.662 (0.049)	0.699 (0.056)
	DoT-Net	0.911 (0.040)	0.878 (0.032)	0.934 (0.038)
	F1DCNN	0.828 (0.026)	0.794 (0.025)	0.855 (0.038)
	FFN	0.727 (0.027)	0.751 (0.018)	0.742 (0.013)
	CNN	0.639 (0.025)	0.582 (0.055)	0.674 (0.041)

Line-diag.	**GSVM**	0.735 (0.026)	0.742 (0.025)	0.749 (0.029)
	HOGMLP	0.756 (0.033)	0.781 (0.023)	0.747 (0.038)
	DoT-Net	0.934 (0.013)	0.926 (0.019)	0.956 (0.025)

TABLE 6: Performance with tile images in multiclass classification

Method	Accuracy	F1 score	AUC
F1DCNN	0.881 (0.022)	0.868 (0.028)	0.943 (0.024)
FFN	0.790 (0.046)	0.685 (0.017)	0.778 (0.043)
CNN	0.848 (0.027)	0.732 (0.043)	0.882 (0.016)
DoT-Net	0.940 (0.019)	0.876 (0.009)	0.976 (0.012)

TABLE 7: Performance with block images in multiclass classification

Method	Accuracy	F1 score	AUC
F1DCNN	0.842 (0.019)	0.832 (0.013)	0.874 (0.023)
FFN	0.532 (0.035)	0.451 (0.027)	0.593 (0.041)
CNN	0.681 (0.024)	0.643 (0.013)	0.716 (0.029)
GSVM	0.449 (0.004)	0.394 (0.007)	0.512 (0.009)
HOGMLP	0.491 (0.019)	0.371 (0.005)	0.532 (0.019)
DoT-Net	**0.941 (0.021)**	**0.929 (0.011)**	**0.952 (0.017)**

CHAPTER V

Knowledge extraction

Knowledge extraction from OCR documents is important to improve the document search and to build the Q&A systems. A good number of industries depends on OCR documents so efficient knowledge extraction improves the value of business, making knowledge extraction an important research topic.

5.1 Related works for knowledge extraction

Existing knowledge extraction technique can be bifolded 1) grammar analysis 2) rule-based analysis. In rule based key word matching systems, knowledge extraction is done by matching texts in documents to a user-defined keyword or key phrase [6, 8, 18, 30]. Text are tokenized by a single space character or a line character to match with the user keywords. This algorithm considers that, all text characters (words) in the document are independent. The performance of this algorithms relies on the user-define keywords.

Ability to Detect relation between two words or sentences, which have a good probability of occurring together (synonyms or regular phrases) or have a close relationship, will have an impact in knowledge extraction. Knowledge extraction from grammar analysis [9, 15, 24, 26, 31] is an algorithm developed to find the relation between the words. The general grammar rules are used as rules for this algorithm. Challenges of this algorithm is it limits to finding the relation between verb-adjective or a noun-verb which follows grammar rules, but cases such as two closely related nouns or verbs are not

considered, for example, "price" and "payment" are nouns, which are often reference together in legal documents, but their relation is not considered by this algorithm.

Another Rule based algorithm which relies on regular expression matching is proposed to obtain the relations between two closely related words is proposed. A set of pre-defined rules are used to match multiple keywords. However, this algorithm limits to documents which follow the rules and requires domain experts to define rules. It is expensive and difficult to generalize the solutions.

In this chapter, we propose an algorithm, which finds the relation between the words by using Query expansion (QE) technique. Knowledge extraction is done by using vector space model and hierarchical document analysis (Dictionary format document achieved in previous chapter).

5.2 Knowledge extraction framework.

We proposed knowledge extraction algorithm to extract the relevant texts for given query, from a corpus of OCR documents. The proposed algorithm includes two phases. In the first phase, an OCR document is reconstructed to structured hierarchical dictionary format as discussed in the chapter 2. In the second phase, the reconstructed documents are further preprocessed. Preprocessing includes removing stop words and special characters, and case folding. Text after preprocessing is tokenized using N-grams and appended into bag of words (BOG).

Concurrently, when a user provides a query, the query is updated by using the query expansion method. hierarchically structured data and the expanded query are then converted to a vector space model (VSM) and compared for comparing the relevance of text entity to the query. Figure 5.1 illustrates the flow diagram of the framework.

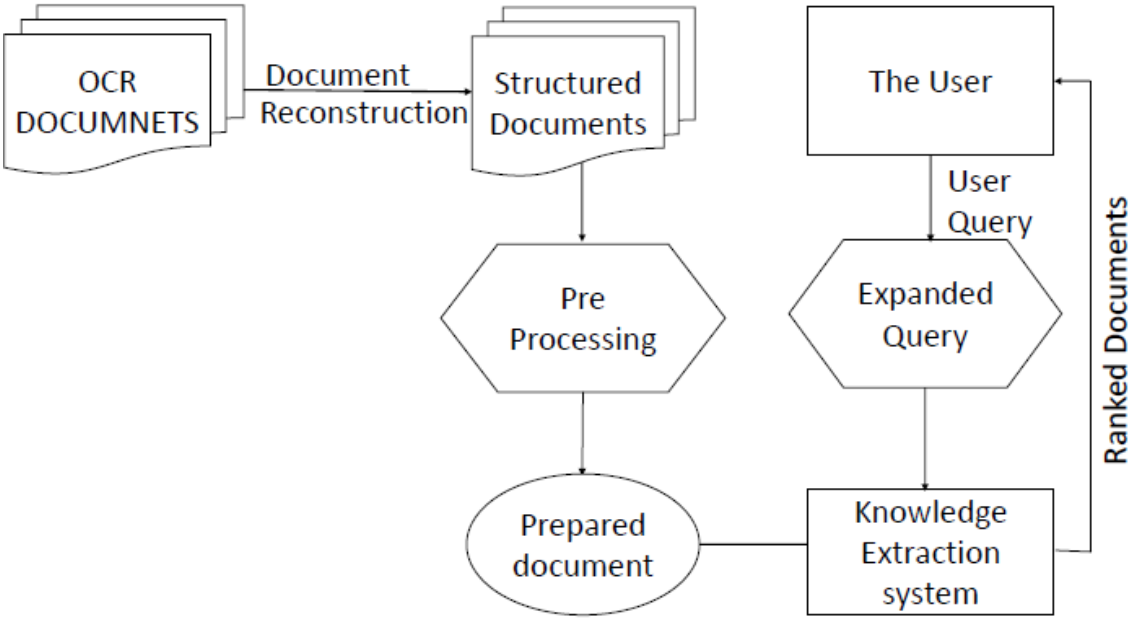


Figure 5.1: Flow diagram of knowledge extraction algorithm

5.2.1 key technologies and terminologies used.

- **Query Expansion:** Number of existing approaches compares the relevance based on a short query, leading to term mismatch retrieval of the information. A short query may lacks the sufficient words necessary to represent the accurate information during the information retrieval. Query Expansion (QE) technique used to address this problem. QE technique often adds new tokens (words) to existing keyword tokens, upon searching terms to generate expanded queries. The

QE technique used in this algorithm is Local analysis technique [29]. In this technique, the top ranked documents assumed to be answer to the answers of the relevant query are tokenized and, appending to the query. The top K frequent terms in newly appended query are considered as expanded query. Relevance feedback [7], can improve the QE technique.

- **Vector Space Model (VSM):** Vector space model is a technique to represent the text entities as vectors. This is most widely used technique in information retrieval [8].

- **Term frequency and Inverse Document frequency (TF-IDF):** Term frequency (TF) is defined as ratio of word occurring in a document to the total words in the document. Inverse document frequency (IDF) is defined as ratio of number of documents to the number of documents which a word repeat.TF*IDF gives the TFIDF scores for the text entity.

- **Cosine Similarity:** Cosine similarity is the measure of similarity between two vectors. In our algorithm, we represent text as vector with their respective TFIDF scores.

5.2.2 Knowledge extraction.

Our algorithm aims at reformulating the query to extract the relevant knowledge from OCR documents for the user defined query. Tokens are collected by using N-grams

technique (sequence of words in length N) [23] from documents and appended in bag of words (BoW) representation. Then, TF-IDF [12, 14] score is computed for the tokens. Each section of the documents is transformed into a vector space, with the values of TF-IDF scores. Cosine similarity [22], technique is used to measure the hierarchical relevance between the expanded query and sections/subsections.

N-grams (N <= 3) is used to create new tokens from the preprocessed sections, new tokens are included with uni-grams in a Bag of words [26]. Initially, most relevant sections to a given query were provided from M documents manually. Preprocessing is performed on the sections. The expanded query is then generated from the BoW by selecting T most frequent of tokens where T is defined by the user. Figure 5.2 illustrates the flow diagram of developing expanded query

The hierarchal structured documents after the document reconstruction process is used for further analysis. Every token in the hierarchal structured document is preprocessed and then assigned values using TF-IDF technique. In the process, each section of a document and the given query are transformed into vectors. Similarity score between the query and all sections is calculated in a pairwise manner using VSM. All of the sections are ranked by the similarity scores and the section that contains the highest similarity score is considered as the most relevant knowledge to the query. If the obtained section does not contain any subsection, the relevant section is retrieved as an output (the most relevant knowledge) for the query. If it contains subsections, the algorithm iterates over the section and determined the most relevant subsection. If the retrieved subsection

contains hierarchical sub-subsection, the method repeats the process and investigates the most relevant subsubsection and so-on.

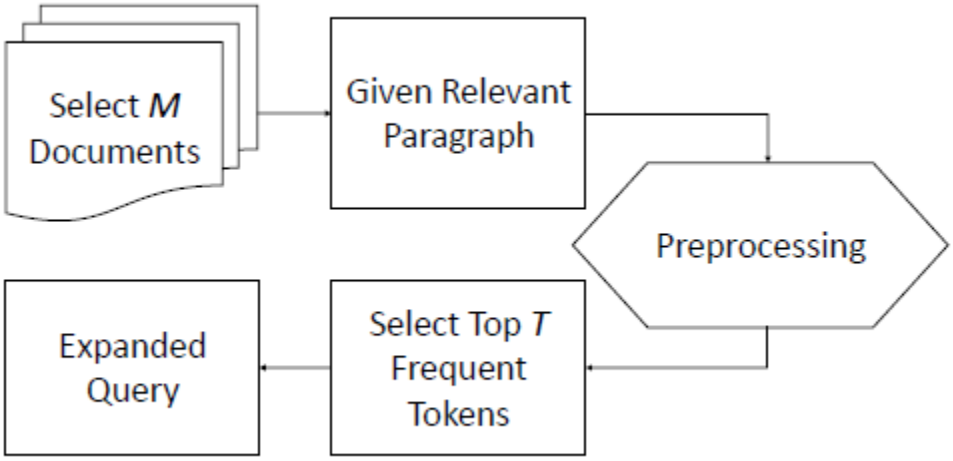


Figure 5.2: Flow diagram of query expansion algorithm

5.3 Results and discussion.

We applied our algorithm to a corpus of OCR documents provided by GE Power (GEP). These documents contain multiple sections such as Appendix, Sections and Exhibits. These, in turn could comprise of multiple layer of subsections. GEP provided keywords (queries) for knowledge extraction. For each query, a ground truth of 5 OCR documents are provided (answers to the given query) from. With this ground truth expanded query is formulated. Example of expanded query is given shown in figure 5.3. We performed data preprocessing since stop words contain less importance and these are common in all documents, these words are filtered out using stop words list of python toolkit, NLTK [5].

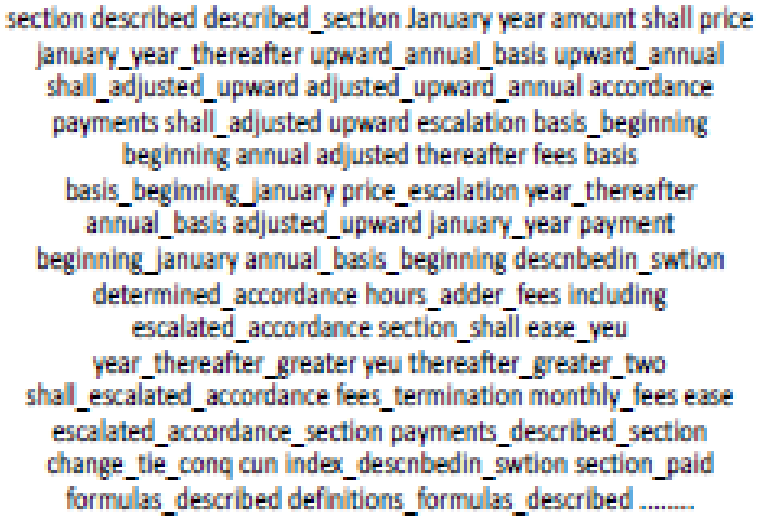


Figure 5.3: Example of expanded query

Our method aims at extracting the most relevant information regarding a query term that a user defines. Specifically, we demonstrate the process with the query term "Liquidated Damages" ("Liquidated Damages" query provides information related to liabilities of industries in case of damages). First, we expanded "Liquidated Damages" query, by using query expansion technique. Figure 5.3 shows the expanded query of "Liquidated Damages". The expanded query is compared with sections in document by VSM. Table 8 shows the top five relevant sections in the document along with relevance scores for the given query "Liquidated damages". "PART 6" shows the highest similarity score of 0.0321, which is the most relevant section to the query.

Table 8: Relevance Ranking of Sections

Sections	Similarity Score
PART 6	0.0321
PART 5	0.0195
PART 9	0.0090
PART 7	0.0032
PREAMBLE	0.0020

The next step is to extract the most relevant portion within Section "PART 6" to query. We compared the expanded query of “Liquidated Damages" to the subsections within Section "PART 6". Table 9 shows the top five most relevant subsections in Section "PART 6". "6.2 Periodic Payments" is with the highest similarity score of 0.03, which is the most relevant subsection to the query. If the section "6.2 Periodic Payments" does not contain any subsections, "6.2 Periodic Payments" is extracted as the most relevant section for given query "Liquidated Damages". If "6.2 Periodic Payments" contains subsections with in it, we extract the most relevant section within Section "6.2 Periodic Payments". We compared the expanded query of the "Liquidated Damages" with subsections within "6.2 Periodic Payments". Table 10 represents the most relevant subsections within "6.2 Periodic Payments" along with the relevance score for the query. "6.2.4 Liquidated Damages Bonus" with the relevance score of "0.2776" is the most relevant section within "6.2 Periodic payments" to the query "Liquidated Damages". Figure 5.4 represent the

most relevant section within the document for the given query term "Liquidated Damages".

Table 9: Relevance Ranking of Sub Sections

Subsections	Similarity Score
6.2 Periodic Payments	0.03
6.12 Initial Spare Parts	0.0
6.4 Extra Work	0.0
6.3 Unplanned Extra Work	0.0
6.26	0.0

Table 10: Relevance Ranking of Sub Sections

Sub-subsections	Similarity Score
6.2.4 Liquidated Damages or Bonus	0.2776
6.2.1 Fixed Lump Sum Annual Payments	0.0
6.2.2 Periodic Price Escalation	0.0
6.2.3 Option for Second Major	0.0

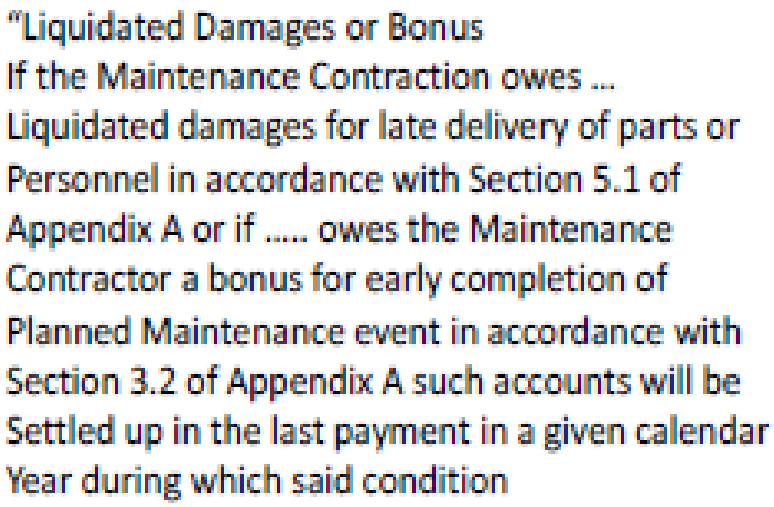


Figure 5.4: Most relevant paragraph to the query

CHAPTER VI

Conclusion

In this study, we proposed a Q&A framework form OCR documents for given user-specific questions with VSM and query expansion techniques. Our framework takes advantage of the proposed novel document analysis algorithms (DLA). Specifically, proposed machine learning algorithms for DLA has outperformed state of the art algorithms in their class. The Proposed texture-based deep learning network (DoT-Net) learns texture features by using dilated convolutional layers. Dilated convolutional layers followed by a max-pooling layer enable one to capture texture features for a classification problem, whereas most dilated convolutional layers have been directly used as a deconvolutional layer. DoT-Net outperformed state of the art DLA algorithms by 10%. To our best knowledge, the proposed image-based machine learning approach for TOC recognition is the first work to utilize image-based machine learning for TOC recognition. Proposed TOC recognition method outperformed existing methods by 6%. Overall proposed Q&A framework could be used for various applications such as automatic knowledge management and document search systems. Our Q&A system has applied to extract domain-specific information from business contracts at GE Power.

Reference

[1] Smith MH. Move over, Manual Processes: E-Document Processing Improves Compliance and Increases Efficiency. Journal of State Taxation. 2011;(Issue 4):41.

[2] Boundary Objects, Agents, and Organizations: Lessons from E-Document System Development in Thailand. 2012 45th Hawaii International Conference on System Sciences, System Science (HICSS), 2012 45th Hawaii International Conference on. 2012:2249. doi:10.1109/HICSS.2012.133.

[3] H. Bast and C. Korzen. 2017. A Benchmark and Evaluation for Text Extraction from PDF. In 2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL) . 1–10. https://doi.org/10.1109/JCDL.2017.7991564

[4] Sanyam Bharara, Sai Sabitha, and Abhay Bansal. 2017. Application of learning analytics using clustering data Mining for Students' disposition analysis. Education and Information Technologies (2017). https://doi.org/10.1007/s10639-017-9645-7

[5] Steven Bird and Edward Loper. 2004. NLTK: The Natural Language Toolkit. In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics . https://doi.org/10.3115/1118108.1118117 arXiv:cs/0205028

[6] R. Chaniago and M. L. Khodra. 2017. Information extraction on novel text using machine learning and rule-based system. In 2017 International Conference on Innovative and Creative Information Technology (ICITech) . 1–6. https://doi.org/10.1109/INNOCIT.2017.8319148

[7] Hang Cui, Ji-Rong Wen, Jian-Yun Nie, and Wei-Ying Ma. 2002. Probabilistic query expansion using query logs. Proceedings of the 11th international conference on World Wide Web (2002). https://doi.org/10.1145/511446.511489

[8] P. M. Darshan. 2017. Ontology based information extraction from resume. In 2017 International Conference on Trends in Electronics and Informatics (ICEI) . 43–47. https://doi.org/10.1109/ICOEI.2017.8300962

[9] T. Erekhinskaya, M. Balakrishna, M. Tatu, S. Werner, and D. Moldovan. 2016. Knowledge extraction for literature review. In 2016 IEEE/ACM Joint Conference on Digital Libraries (JCDL) . 221–222.

[10] J. Fan, A. Kalyanpur, D. C. Gondek, and D. A. Ferrucci. 2012. Automatic knowledge extraction from documents. IBM Journal of Research and Development (2012). https://doi.org/10.1147/JRD.2012.2186519

[11] Alexander Gelbukh. [n. d.]. Natural Language Processing. ([n. d.]), 7695. https://doi.org/10.1109/ICHIS.2005.79

[12] Vishal Gupta and Gurpreet S. Lehal. 2009. A survey of text mining techniques and applications. https://doi.org/10.4304/jetwi.1.1.60-76

[13] M. Hanumanthappa and Deepa T. Nagalavi. 2015. Identification and extraction of different objects and its location from a Pdf file using efficient information retrieval tools. In Proceedings of the IEEE International Conference on Soft-Computing and Network Security, ICSNS 2015 . https://doi.org/10.1109/ICSNS.2015.7292375

[14] Siham Jabri, Azzeddine Dahbi, Taoufiq Gadi, and Abdelhak Bassir. 2018. Ranking of text documents using TF-IDF weighting and association rules mining. In Proceedings of the 2018 International Conference on Optimization and Applications, ICOA 2018 . https://doi.org/10.1109/ICOA.2018.8370597

[15] A. Kanev, S. Cunningham, and T. Valery. 2017. Application of formal grammar in text mining and construction of an ontology. In 2017 Internet Technologies and Applications (ITA) . 53–57. https://doi.org/10.1109/ITECHA.2017.8101910

[16] R. Kumaravel, S. Selvaraj, and M. C. 2018. A Multi-Domain Layered Approach in Development of Industrial Ontology to Support Domain Identification for Unstructured Text. IEEE Transactions on Industrial Informatics (2018), 1–1. https: //doi.org/10.1109/TII.2018.2835567

[17] David D. Lewis. 1992. An Evaluation of Phrasal and Clustered Representations on a Text Categorization Task. In Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval (1992). https://doi.org/10.1145/133160.133172

[18] Ahsan Mahmood, Hikmat Ullah Khan, Zahoor-Ur-Rehman, and Wahab Khan. 2018. Query based information retrieval and knowledge extraction using Hadith datasets. In Proceedings - 2017 13th International Conference on Emerging Technologies, ICET2017 . https://doi.org/10.1109/ICET.2017.8281714

[19] Andres McCallum and Kamal Nigam. 1998. A Comparison of Event Models for Naive Bayes Text Classification. AAAI/ICML-98 Workshop on Learning for Text Categorization (1998). https://doi.org/10.1.1.46.1529 arXiv:0-387-31073-8

[20] Thi Tuyet Hai Nguyen, Antoine Doucet, and Mickael Coustaty. 2018. En-hancing Table of Contents Extraction by System Aggregation. In Proceedings of the International Conference on Document Analysis and Recognition, ICDAR. https://doi.org/10.1109/ICDAR.2017.48

[21] Gerard Salton and Chris Buckley. 1990. Improving retrieval performance by relevance feedback. Journal of the American Society for Information Science (1990). https://doi.org/10.1002/(SICI)1097-4571(199006)41:4<288::AID-ASI8>3.0.CO;2-H arXiv:arXiv:1011.1669v3

[22] M Steinbach, G Karypis, and V Kumar. 2000. A Comparison of Document Clustering Techniques. KDD workshop on text mining (2000). https://doi.org/10.1109/ICCCYB.2008.4721382

[23] Chade Meng Tan, Yuan Fang Wang, and Chan Do Lee. 2002. The use of bigrams to enhance text categorization. Information Processing and Management (2002). https://doi.org/10.1016/S0306-4573(01)00045-0

[24] R. Upadhyay and A. Fujii. 2016. Semantic knowledge extraction from research documents. In 2016 Federated Conference on Computer Science and Information Systems (FedCSIS) . 439–445.

[25] D. G. Vasques, A. C. Zambon, G. B. Baioco, and P. S. Martins. 2016. An Approach to Knowledge Acquisition Based on Verbal Semantics. In 2016 49th Hawaii International Conference on System Sciences (HICSS) . 4144–4153. https: //doi.org/10.1109/HICSS.2016.514

[26] Hanna M. Wallach. 2006. Topic Modeling: Beyond Bag-of-words. In Proceedings of the 23rd International Conference on Machine Learning (ICML '06) . ACM, New York, NY, USA, 977–984. https://doi.org/10.1145/1143844.1143967

[27] Ho Chung Wu, Robert Wing Pong Luk, Kam Fai Wong, and Kui Lam Kwok. 2008. Interpreting TF-IDF term weights as making relevance decisions. ACM Transactions on Information Systems (2008). https://doi.org/10.1145/1361684.1361686

[28] Zhaohui Wu, Prasenjit Mitra, and C. Lee Giles. 2013. Table of contents recognition and extraction for heterogeneous book documents. In Proceedings of the International Conference on Document Analysis and Recognition, ICDAR . https://doi.org/10.1109/ICDAR.2013.244

[29] Jinxi Xu and W Bruce Croft. 1996. Query expansion using local and global document analysis. Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '96 (1996). https://doi.org/10.1145/243199.243202

[30] Wenhao Zhu, Laihu Luo, Chaoyou Ju, and Bofeng Zhang. 2012. Cross language information extraction for digitized textbooks of specific domains. In Proceedings - 2012 IEEE 12th International Conference on Computer and Information echnology, CIT 2012 . https://doi.org/10.1109/CIT.2012.226

[31] S. T. Zuhori, M. A. Zaman, and F. Mahmud. 2017. Ontological knowledge extraction from natural language text. In 2017 20th International Conference of Computer and Information Technology (ICCIT) . 1–6. https://doi.org/10.1109/ICCITECHN.2017.8281776

[32] A. Belaid, L. Pienon, and N. Valverde. 2000. Part-of-Speech Tagging for Table of Contents Recognition. Proceedings 15th International Conference on Pattern Recognition. ICPR-2000 4 (2000), pp. 1–4. https://doi.org/10.1109/ICPR.2000.902955

[33] F. Le Bourgeois, H. Emptoz, and S. S. Bensafi. 2001. Document Understanding using Probabilistic Relaxation: Application on Tables of Contents of Periodicals. (Sep. 2001), pp. 508–512. https://doi.org/10.1109/ICDAR.2001.953841

[34] H. Déjean and J. Meunier. 2005. Structuring Documents According to Their Table of Contents. (2005), pp. 2–9. https://doi.org/10.1145/1096601.1096605

[35] L. Gao, Z. Tang, X. Lin, X. Tao, and Y. Chu. 2009. Analysis of Book Document's Table of Content based on Clustering. (July 2009), pp. 911–915. https://doi.org/10.1109/ICDAR.2009.143

[36] Y. Jayabal, C. Ramanathan, and M. Sheth. 2012. Challenges in Generating Bookmarks from TOC Entries in e-Books. (2012), pp. 37–40. https://doi.org/10.1145/2361354.2361363

[37] S. Mandal, S.P. Chowdhury, A.K. Das, and B. Chanda. 2003. Automated Detection and Segmentation of Table of Contents Page from Document Images. Proceedings of the International Conference on Document Analysis and Recognition, ICDAR 2003-January (2003), pp. 398–402. https://doi.org/10.1109/ICDAR.2003.1227697

[38] S. Marinai, E. Marino, and G. Soda. 2010. Table of Contents Recognition for Converting PDF Documents in e-Book Formats. (2010), pp. 73–76. https://doi.org/10.1145/1860559.1860576

[39] R. Parikh and A. Vasant. 2013. Table of Content Detection using Machine Learning: Proposed System. International Journal of Artificial Intelligence & Applications (IJAIA) 4, 3 (2013), pp. 13–21. https://doi.org/10.5121/ijaia.2013.4302

[40] P. Sarkar and E. Saund. 2008. On the Reading of Tables of Contents. (Sep. 2008), pp. 386–393. https://doi.org/10.1109/DAS.2008.87

[41] Z. Wu, P. Mitra, and C. L. Giles. 2013. Table of contents recognition and extraction for heterogeneous book documents. In 2013 12th International Conference on Document Analysis and Recognition . pp. 1205–1209. https://doi.org/10.1109/ICDAR.2013.244

[42] S. Bhowmik, R. Sarkar, M. Nasipuri, and D. Doermann, "Text andnon-text separation in offline document images: a survey,"InternationalJournal on Document Analysis and Recognition (IJDAR), vol. 21, no. 1,pp. 1–20, Jun 2018. [Online]. Available:https://doi.org/10.1007/s10032-018-0296-z

[43] A. Suvichakorn, S. Watcharabusaracum, and W. Sinthupinyo, “Simplelayout segmentation of gray-scale document images,” inInternationalWorkshop on Document Analysis Systems. Springer, 2002, pp. 245–248.

[44] T. Guan and H. Zhu, “Atrous faster r-cnn for small scale objectdetection,” in2017 2nd International Conference on Multimedia andImage Processing (ICMIP), March 2017, pp. 16–21.

[45] Jin Wu, Wu-Mo Pan, Jian-Ming Jin, and Qing-Ren Wang, “Performanceevaluation and benchmarking on document layout analysis algorithms,”2004.

[46] M. Diem, F. Kleber, and R. Sablatnig, “Text classification and documentlayout analysis of paper fragments,” inProceedings of the InternationalConference on Document Analysis and Recognition, ICDAR, 2011.

[47] A. K. Sah, S. Bhowmik, S. Malakar, R. Sarkar, E. Kavallieratou, andN. Vasilopoulos, “Text and non-text recognition using modified hogdescriptor,” in2017 IEEE Calcutta Conference (CALCON), Dec 2017,pp. 64–68.

[48] V. P. Le, N. Nayef, M. Visani, J. M. Ogier, and C. D. Tran, “Textand non-text segmentation based on connected component features,” inProceedings of the International Conference on Document Analysis andRecognition, ICDAR, 2015.

[49] O. K. Oyedotun and A. Khashman, “Document segmentation using tex-tural features summarization and feedforward neural network,”AppliedIntelligence, 2016.

[50] M. P. Viana and D. A. B. Oliveira, “Fast CNN-Based Document LayoutAnalysis,” inProceedings - 2017 IEEE International Conference onComputer Vision Workshops, ICCVW 2017, 2018.

[51] L. Kang, J. Kumar, P. Ye, Y. Li, and D. Doermann, “Convolutional neuralnetworks for document image classification,” in2014 22nd International Conference on Pattern Recognition. IEEE, 2014, pp. 3168–3172.

[52] A. W. Harley, A. Ufkes, and K. G. Derpanis, “Evaluation of deepconvolutional nets for document image classification and retrieval,” inProceedings of the International Conference on Document Analysis andRecognition, ICDAR, 2015.

[53] E. Crestan and P. Pantel, “Web-scale knowledge extraction from semi-structured tables,” 2010.

[54] E. Anisimova, P. Páta, and M. Blažek, “Stellar Object Detection Usingthe Wavelet Transform,”Acta Polytechnica, vol. 51, no. 6, p. 9, 2011.

[55] A. Constantin, J. Ding, and Y. Lee, “Accurate road detection fromsatellite images using modified u-net,” in2018 IEEE Asia PacificConference on Circuits and Systems (APCCAS), Oct 2018, pp. 423–426.

[56] E. Anisimova, J. Bednář, and P. Páta, “Efficiency of wavelet coefficientsthresholding techniques used for multimedia and astronomical imagedenoising,” in2013 International Conference on Applied Electronics,Sep. 2013, pp. 1–4.

[57] Z. Hu, T. Turki, N. Phan, and J. T. L. Wang, “A 3d atrous convolutionallong short-term memory network for background subtraction,”IEEEAccess, vol. 6, pp. 43 450–43 459, 2018.

[58] Y. Liu, X. Zhu, X. Zhao, and Y. Cao, “Adversarial learning forconstrained image splicing detection and localization based on atrousconvolution,”IEEE Transactions on Information Forensics and Security,pp. 1–1, 2019.

[59] N. Jin and Z. Long, “Effusion area segmentation for knee joint ultrasoundimage based on atrous-fcn with snake model algorithm,” in2018 11thInternational Congress on Image and Signal Processing, BioMedicalEngineering and Informatics (CISP-BMEI), Oct 2018, pp. 1–9.

[60] Z. Feng, H. Yong, and S. Xukun, “Granet: Global refinement atrousconvolutional neural network for semantic scene segmentation,” in201825th IEEE International Conference on Image Processing (ICIP), Oct2018, pp. 1568–1572.

[61] Y. Liu and M. D. Levine, “Multi-path region-based convolutional neuralnetwork for accurate detection of unconstrained "hard faces",” in201714th Conference on Computer and Robot Vision (CRV), May 2017, pp.183–190.

[62] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille,“Deeplab: Semantic image segmentation with deep convolutional nets,atrous convolution, and fully connected crfs,”IEEE Transactions onPattern Analysis and Machine Intelligence, vol. 40, no. 4, pp. 834–848,April 2018.

[63] A. Antonacopoulos, S. Pletschacher, D. Bridson, and C. Papadopoulos,“ICDAR2009 page segmentation competition,” inProceedings of the In-ternational Conference on Document Analysis and Recognition, ICDAR,2009.

[64] N. Yi, C. Li, X. Feng, and M. Shi, “Research and improvement ofconvolutional neural network,” in2018 IEEE/ACIS 17th InternationalConference on Computer and Information Science (ICIS), June 2018,pp. 637–640.

[65] V. Singh and B. Kumar, “Document layout analysis for Indian news-papers using contour based symbiotic approach,” in2014 InternationalConference on Computer Communication and Informatics: Ushering inTechnologies of Tomorrow, Today, ICCCI 2014, 2014.